

IST. EL. INF.
BIBLIOTECA
Posiz. **Archivio**

BO-06

1986

RECUPERO E CONCORRENZA NELLE BASI DI DATI

di Elvira Locuratolo

Estratto da: SISTEMI E AUTOMAZIONE, xxxii, n. 271 (1986), 677-680

01/11/2010

RECUPERO E CONCORRENZA NELLE BASI DI DATI

di Elvira Locuratolo (*)

Una tecnica per la soluzione di alcuni problemi di concorrenza basata sullo schema di Lorie

1. Introduzione

Durante gli ultimi dieci anni sono stati proposti numerosi meccanismi per il recupero dei dati e il controllo della concorrenza nella basi di dati; tuttavia, la maggior parte di essi affronta i due problemi in modo indipendente ignorando che reciproche interazioni spingono a trattamenti risolutivi unificati.

Il "recupero dei dati" consiste nel ripristinare lo stato del sistema qualora un malfunzionamento produca inconsistenze nei dati, o altre forme di danneggiamento; mentre il "controllo della concorrenza" consiste nell'impedire l'insorgere di inconsistenze dovute al contemporaneo accesso di più utenti agli stessi dati.

Invero, se un dato "x" deve essere contemporaneamente letto dall'utente A ed aggiornato all'utente B, l'argomento della let-

tura dipende dall'ordine relativo delle operazioni (*conflitto di tipo read-write*); ed analogamente, se un dato "x" deve essere contemporaneamente aggiornato dagli utenti A e B, la sua determinazione finale dipende dall'ordine relativo delle scritture (*conflitto di tipo write-write*).

Per evidenziare le connessioni tra i due problemi, il presente articolo mostra come una tecnica di sicurezza proposta dal Lorie [5], oltre ad essere vantaggiosamente utilizzata per il recupero dei dati, può esserlo anche per la soluzione di alcuni problemi riguardanti il controllo della concorrenza. A tale scopo, si presenta una tecnica di sincronizzazione rw e si descri-

ve una sua implementazione che generalizza lo schema di Lorie [4]. La tecnica proposta scompone le transazioni in fasi opportune e le elabora secondo modalità che non comportano situazioni di stallo, rinvii indefiniti o restart ciclici.

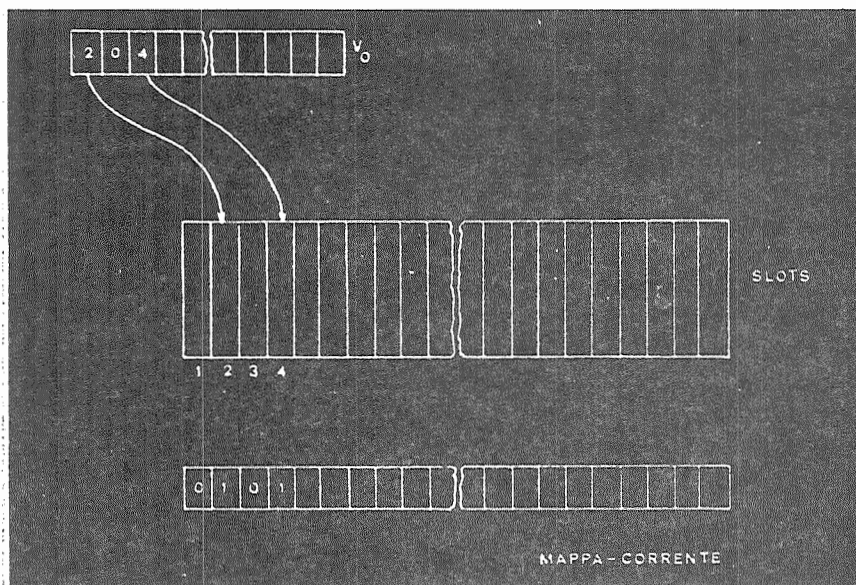
Il lavoro è articolato nelle seguenti sezioni:

- la tecnica di Lorie;
- tecniche per il controllo della concorrenza;
- un meccanismo di sincronizzazione rw;
- ampliamento dello schema di Lorie;
- conclusioni.

2. La tecnica di Lorie

Per tutelare l'integrità dei dati dai possibili malfunzionamenti, la tecnica di Lorie opera in modo da mantenere lo stato consistente dei dati S_k durante l'intervallo di tempo $(t, t + \Delta t)$; costruire parallelamente lo stato S_{k+1} ed eseguire una operazione "atomica" al tempo $t + \Delta t$ per generare il nuovo stato consistente S_{k+1} e rilasciare S_k . Un'operazione atomica permette o di eseguire tutte le scritture per passare al nuovo stato consistente, o di non eseguirne alcuna per ripristinare l'integrità nei casi di malfunzionamento. Per raggiungere tali scopi, Lorie

Figura 1



(*) La dott.ssa Elvira Locuratolo opera presso l'Istituto di Elaborazione delle Informazioni, CNR, Pisa.

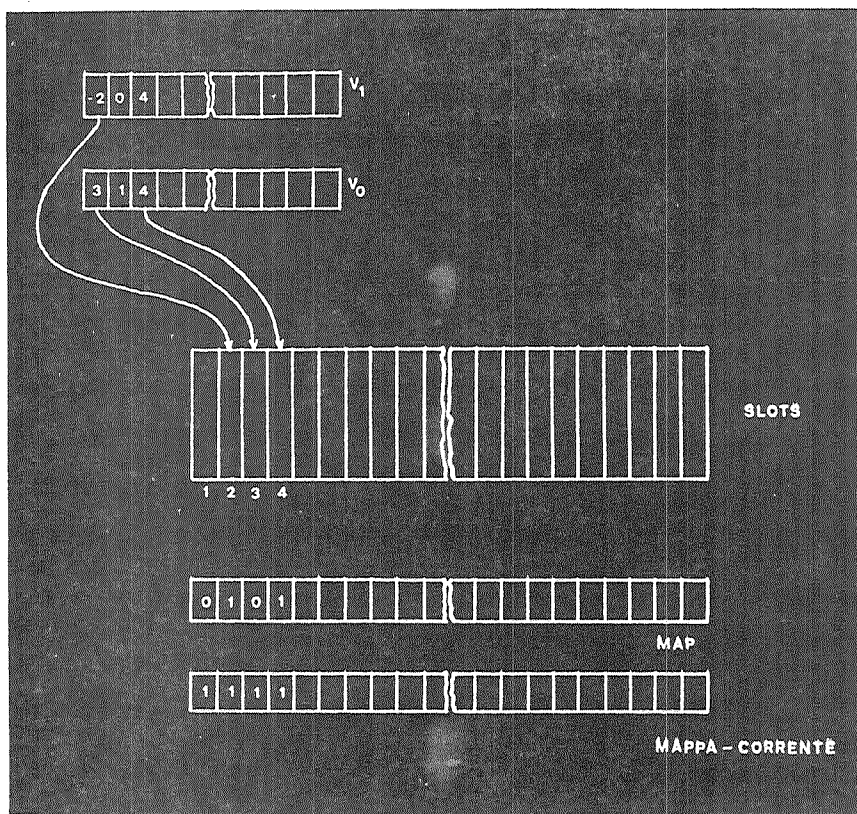


Figura 2

amplia lo schema di gestione della memoria rappresentato in Figura 1. In esso il generico dato "x" è allocato nello slot "y", la corrispondenza tra x e y è mantenuta nel vettore V_0 da cui risulta $V_0(x) = y$ e lo stato degli slot è rappresentato da un vettore di bit MAPPA-CORRENTE definito dalle seguenti posizioni: MAPPA-CORRENTE (y) = 1 se lo slot y è occupato; MAPPA-CORRENTE (y) = 0 se lo slot y è libero.

Lo schema proposto dal Lorie, invece, è rappresentato in Figura 2. Esso consiste nel mantenere oltre alla copia di vettori (V_0 , MAPPA-CORRENTE) caratterizzante lo stato corrente dei dati S_{k+1} , anche la copia di vettori (V_1 , MAP) caratterizzante lo stato consistente S_k . I vettori V_1 e MAP hanno rispettivamente le dimensioni e le caratteristiche di V_0 e MAPPA-CORRENTE e sono mantenuti sul disco, mentre i vettori V_0 e MAPPA-CORRENTE sono mantenuti in memoria centrale.

Premesso ciò, l'aggiornamento

del dato "x" nell'intervallo di tempo $(t, t + \Delta t)$ è eseguito allocando la nuova copia del dato "x" in uno slot disponibile y' e mantenendo la vecchia copia nello slot originale y; tale aggiornamento deve riflettersi anche sui vettori V_0 e MAPPA-CORRENTE che rappresentano lo stato corrente dei dati, ma non sui vettori V_1 e MAP che rappresentano lo stato consistente. Pertanto, nel tempo $(t, t + \Delta t)$ i vettori in memoria centrale e sul disco assumono i seguenti rispettivi valori:

$V_0(x) = y'$
 MAPPA-CORRENTE (y') = 1
 $V_1(x) = -y$ in memoria;
 $V_1(x) = y$
 MAP (y) = 1
 MAP (y') = 0 sul disco.

Al tempo $t + \Delta t$, il passaggio al nuovo stato consistente S_{k+1} è eseguito rilasciando gli slot contenenti le vecchie copie dei dati aggiornati, cioè ponendo MAPPA-CORRENTE (y) = 0, V y per cui risulta $V_1(x) = -y$ in memoria, ed inoltre copiando rispettivamente V_0 in V_1 e MAPPA-CORRENTE in MAP.

Per mantenere integro il conte-

nuto di V_1 e MAP fino al termine del passaggio allo stato S_{k+1} , è opportuno avere sul disco anche la duplicazione di tali vettori.

In caso di malfunzionamento, la retrocessione al precedente stato S_k è eseguita copiando rispettivamente V_1 in V_0 e MAP in MAPPA-CORRENTE.

La tecnica di Lorie elabora dunque le operazioni di passaggio al nuovo stato consistente e retrocessione allo stato precedente non mediante la diretta copiatura dei dati, ma soltanto mediante quella dei loro puntatori. Ne risultano dei costi estremamente vantaggiosi; infatti è stato approvato che con meno di 600 accessi è possibile ottenere il nuovo stato consistente di un database di circa 30.000 dati aggiornato quasi interamente [5] e che tale costo è pari a circa 1/4 di quello richiesto a parità di condizioni dall'esplicita copiatura dei dati [3].

3. Tecniche per il controllo della concorrenza

La maggior parte dei metodi classici per il controllo della concorrenza risulta definito da variazioni e/o composizioni delle due seguenti tecniche basilari:

- 1) Two-Phase Locking (2PL);
- 2) Timestamp ordering (T/O).

Entrambe si basano sul rilevamento dei conflitti e possono essere utilizzate sia per la sincronizzazione rw, ovvero per la risoluzione dei conflitti di tipo read-write, sia per la sincronizzazione ww, ovvero per la risoluzione dei conflitti di tipo write-write.

La 2PL opera nel seguente modo:

- a) le transazioni pongono sul generico dato "x" un blocco di lettura prima di leggerlo e un blocco di scrittura prima di aggiornarlo;
- b) transazioni diverse non pongono simultaneamente blocchi in conflitto;
- c) se una transazione rilascia un

blocco non può acquisirne altri. Per effetto del punto c) l'esecuzione di una transazione consta di due fasi; esse sono dette rispettivamente *growing phase* (o fase di acquisizione blocchi) e *shrinking phase* (o fase di rilascio blocchi); tuttavia, se un blocco non può essere acquisito per effetto del punto b), la transazione che lo ha richiesto è posta in coda di attesa. Dopo che un dato è stato sbloccato, i comandi per quel dato sono elaborati secondo l'ordine FIFO. La 2PL può generare situazioni di stallo dovute ad eventi che bloccano vicendevolmente le transazioni.

La T/O, invece, esegue le operazioni in conflitto in modo consistente ad un prefissato ordinamento delle transazioni; esso è ottenuto associando a ciascuna transazione un valore (*timestamp*) calcolato in funzione del proprio tempo di partenza.

La T/O non comporta situazioni in stallo, ma può generare ripartenze continue di una stessa transazione.

4. Un meccanismo di sincronizzazione RW

La tecnica che si descrive nella presente sezione non controlla l'ordine di esecuzione dei conflitti, ma ne impedisce addirittura l'insorgere.

Essa richiede il contemporaneo mantenimento di uno stato corrente e di uno stato consistente dei dati e si riferisce al caso di aggiornamenti apportati da un solo utente. Il parallelismo riguarda quindi le queries Q_1, Q_2, \dots, Q_n e il processo di modifica M .

Per evitare l'insorgere di conflitti di tipo read-write, la tecnica decompone il processo M in una fase di esecuzione "ME", una fase di attesa "MW" e una fase di copiatura "MC".

La fase "ME" consiste nel costruire lo stato corrente dei dati senza alterare lo stato consistente; la fase "MW" consiste in un blocco per l'esecuzione di M e la

fase "MC" nel passaggio al nuovo stato consistente.

Le tre fasi vengono elaborate secondo le seguenti modalità di parallelismo:

a) durante la "ME", la modifica e le queries procedono in parallelo riferendosi rispettivamente allo stato corrente e allo stato consistente dei dati;

b) durante la "MW", le queries in esecuzione sono portate a termine e quelle non ancora partite sono bloccate;

c) durante la "MC" viene creato il nuovo stato consistente.

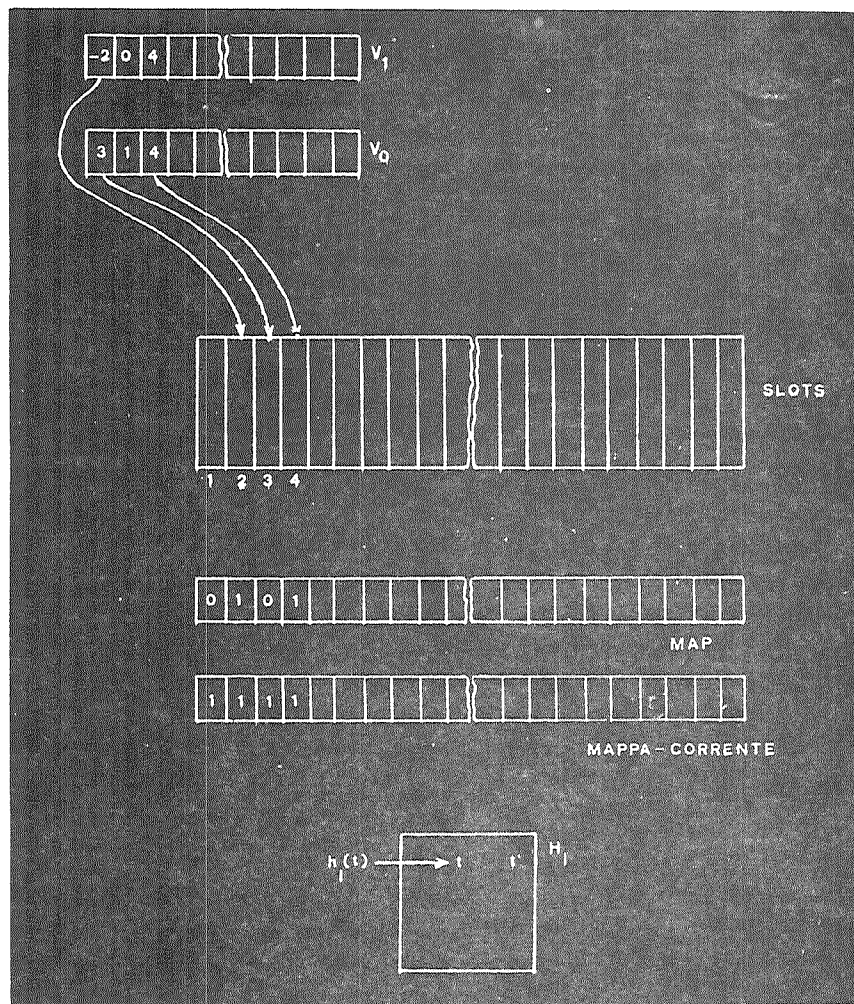
Poiché tale esecuzione risulta computazionalmente equivalente all'esecuzione seriale ($Q_1, \dots, Q_k, M, Q_{k+1}, \dots, Q_n$), con Q_1, \dots, Q_k queries elaborate durante le fasi "ME" ed "MW", la tecnica descritta è corretta; inoltre, se il DBMS adotta la tecnica di Lorie, il mantenimento del duplice

stato dei dati non richiede alcun costo aggiuntivo. Infatti, in tal caso, la fase "ME" si sviluppa durante l'intervallo di tempo $(t, t + \Delta t)$ e l'accesso ai dati avviene tramite il vettore V_0 per il processo M e tramite il vettore V_1 per le queries; la fase "MW" inizia al tempo $t + \Delta t$ e termina a completamento di tutte le queries in esecuzione e la fase "MC" si sviluppa successivamente alla "MW". Infine si osserva che, a differenza dei meccanismi classici, la tecnica descritta non comporta situazioni di stallo o ripartenze continue e non richiede moduli per l'acquisizione e il rilascio di blocchi o timestamp.

5. Ampliamento dello schema di Lorie

Per elevare ulteriormente il parallelismo delle transazioni, il meccanismo proposto può esse-

Figura 3



re generalizzato decomponendo anche la generica query Q_1 nelle due fasi Q_1^1 e Q_1^2 . La fase Q_1^1 termina quando la Q_1 non richiede più dati da leggere per la prima volta; ma fase Q_1^2 è successiva alla Q_1^1 .

Premesso ciò, se si fa in modo che le eventuali letture delle Q_1^2 si riferiscano a dati letti durante le Q_1^1 , la fase "MW" può essere ridotta soltanto fino al completamento delle Q_1^1 e la fase "MC" può essere elaborata parallelamente alle Q_1^2 .

Ciò può essere tecnicamente realizzato ampliando lo schema di Lorie come mostrato in Figura 3; a tale scopo, detto t il generico dato letto per la prima volta da Q_1 e t' il generico slot disponibile del disco, si procede nel seguente modo:

- a) si associa ad ogni query Q_1 una tabella a doppia entrata inizialmente nulla H_1 e una funzione hash h_1 ;
- b) si mantiene la coppia (t, t') nella posizione $h_1(t)$ della H_1 ;
- c) si copia il dato t nello slot t' prima di un eventuale rilascio di t .

Le scritture prodotte dal punto c) abilitano il sistema ad elaborare parallelamente la "MC" e le Q_1 poiché le eventuali letture dei dati t possono essere effettuate negli slot t' . Questi ultimi sono poi rilasciati a completamento delle Q_1 .

6. Conclusioni

Il presente lavoro mostra come la tecnica di Lorie oltre ad essere vantaggiosamente utilizzata per il recupero dei dati può esserlo anche per la soluzione di alcuni problemi riguardanti il controllo della concorrenza. A tale scopo, si descrive una tecnica che impedisce l'insorgere dei conflitti di tipo read-write decomponendo le transazioni in fasi opportune ed eseguendole secondo modalità che non comportano situazioni di stallo e rinvii indefiniti. La tecnica è utilizzabile nei contesti applica-

tivi che non richiedono aggiornamenti immediati e che affidano le modifiche ad un solo utente (anagrafe, biblioteche ecc.); tuttavia, lo studio sull'utilizzo della tecnica di Lorie può essere ampliato per risolvere anche altri problemi della concorrenza. Infine, si rende esplicito che la tecnica di Lorie è applicabile a sistemi operativi che trasformano gli indirizzi degli slot in indirizzi fisici effettivi; tuttavia, tale limitazione può essere facilmente rimossa se si adotta uno schema di allocazione statica dei dati e se si continua a mantenere lo stato consistente e lo stato corrente nella stessa rappresentazione. Per realizzare ciò, basta sostituire ai vettori V_0 e V_1 una tabella a doppia entrata H in modo tale che, detto z un dato da modificare e z' una locazione disponibile del disco, la corrispondenza tra z e z' sia mantenuta inserendo la coppia (z, z') nella H ed inoltre la nuova copia del dato sia inserita nella locazione z' mentre la copia originale rimanga inalterata nella locazione z .

Ringraziamenti. Ringrazio il prof. R. Sprugnoli del Dipartimento di Scienze dell'Informazione dell'Università di Pisa per le discussioni sull'argomento trattato.

BIBLIOGRAFIA

- [1] Bernstein P.A. and Goodman N., "Fundamental Algorithms for Concurrency Control in a Distributed Database System" Techic. rep. CCA-80-05.
- [2] Bernstein P.A. and Shipman D., "The Correctness of Concurrency Mechanisms for Distributed Database" ACM Trans. on Database Systems, Vol. 5, No 1, March 1980.
- [3] Locuratolo E., "Tecniche di Sicurezza nelle Basi di Dati" Atti del Congresso Annuale AICA '80, Vol. 2, pp. 1488-1500.
- [4] Locuratolo E., "Recupero dei Dati e Controllo della Concorrenza nei DBMS" Atti del Congresso Annuale AICA '85, Vol. 1, pp. 263-278.
- [5] Lorie R.A., "Physical Integrity in a Large Segmented Database" ACM Trans, on Database Systems, Vol. 2, No. 1, March '77, pp. 91-104.
- [6] Wiederhold G., "Database Design" McGraw-Hill, No. 1, 1977.